Mathematical Programming 78 (1997) 1–27

A dual-active-set algorithm for positive semi-definite quadratic programming

N.L. Boland *

Department of Mathematics, University of Melbourne, Parkville, VIC 3052, Australia [1]

Received 1 February 1993; revised manuscript received 12 March 1996

Abstract

Because of the many important applications of quadratic programming, fast and efficient methods for solving quadratic programming problems are valued. Goldfarb and Idnani (1983) describe one such method. Well known to be efficient and numerically stable, the Goldfarb and Idnani method suffers only from the restriction that in its original form it cannot be applied to problems which are positive semi-definite rather than positive definite. In this paper, we present a generalization of the Goldfarb and Idnani method to the positive semi-definite case and prove finite termination of the generalized algorithm. In our generalization, we preserve the spirit of the Goldfarb and Idnani method, and extend their numerically stable implementation in a natural way.

Keywords: Quadratic programming; Positive semi-definite; Convex optimization; Active-set method

1. Introduction

Quadratic programming has a long history and a multitude of application areas. In recent years, quadratic programming methods have become increasingly important because of their relevance to algorithms for solving general nonlinear convex programming problems (see, for example, the methods discussed in [12,9,30,16,17,21–23,26,27], and more recently [19]). Lin and Pang [18] give a thorough survey of quadratic programming methods.

Quadratic programming methods in the class known as active-set methods have had significant success. They are usually more memory efficient than methods of other types

* Supported in part by ATERB, NSERC and the ARC.

[1] Much of this work was done in the Department of Mathematics at the University of Western Australia and in the Department of Combinatorics and Optimization at the University of Waterloo.

0025-5610/97/$17.00
Published by Elsevier Science B.V.
PII S0025-5610(96)00069-X

and often enjoy finite termination. Fletcher [9] describes a number of active-set methods, while Goldfarb and Idnani [14] present a dual-active-set method for strictly convex problems. Their extensive computational experience shows their algorithm to be fast as well as efficient. Powell [24] implemented the Goldfarb and Idnani method, including an extension to cope with ill-conditioned problems, and found it to compare favourably with other approaches [25]. The Goldfarb and Idnani method has some other useful features: it requires no costly first phase to determine a feasible starting point, since it is a dual feasible rather than a primal feasible method; and it is amenable to specialization, as the inverse shortest paths algorithm of Burton and Toint [4] and the active-set-on-a-graph algorithm for network optimization [1–3] and multicommodity flows [10,3] attest. Its importance as a method with wide applicability is confirmed by Stoer [28], who develops a related algorithm for solving linear least squares problem. However like a large proportion of quadratic programming methods, the Goldfarb and Idnani dual-active-set method only solves positive definite problems, whereas in practice, problems may often be positive semi-definite. (Note that in the context of linear least squares problems, Stoer's method does avoid the assumption of strict convexity of the objective function.) In this paper we show how the Goldfarb and Idnani dual-active-set method can be generalized to solve positive semi-definite problems. Our primary purpose is theoretical: our generalization is intended to provide a theoretical basis for specializations, such as that given in [2,3], in which the ability to solve semi-definite problems is highly desirable. However we do give some indication of how issues arising in a practical implementation might be addressed, and describe a numerically stable implementation which is a natural extension of that given by Goldfarb and Idnani.

The original dual-active-set method of Goldfarb and Idnani [14] solves the positive definite quadratic programming problem (PDQPP) described below:

$$\begin{array}{ll} \min & \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}Q\boldsymbol{x}+\boldsymbol{p}^{\mathrm{T}}\boldsymbol{x} \\ \text{subject to} & C^{\mathrm{T}}\boldsymbol{x}\geqslant \boldsymbol{b} \end{array}$$

where Q is a symmetric positive definite matrix. If $\boldsymbol{x}$ is also subject to equality constraints, the algorithm can be modified appropriately without difficulty. For simplicity, we will not include equality constraints here.

The problem we will be able to solve with the generalized method is called the positive semi-definite quadratic programming problem (SDQPP), and is the same as the PDQPP except that Q may now be positive semi-definite. To make the problem easier to handle we will require it to be in a special form:

$$\begin{array}{ll} \min\limits_{(\boldsymbol{x},\,\boldsymbol{y})} & f(\boldsymbol{x},\,\boldsymbol{y}) \stackrel{\text{def}}{=} \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}Q\boldsymbol{x}+\boldsymbol{p}^{\mathrm{T}}\boldsymbol{x}+\boldsymbol{\rho}^{\mathrm{T}}\boldsymbol{y} \\ \text{subject to} & C^{\mathrm{T}}\boldsymbol{x}+D^{\mathrm{T}}\boldsymbol{y}\geqslant \boldsymbol{b} \end{array}$$

where the matrix Q is now positive definite, with dimension equal to the rank of the original matrix. The variables $\boldsymbol{x}$ and $\boldsymbol{y}$ are the partitions of the original variables into "quadratic" and "linear" parts. In a network optimization problem with arc-separable costs, this can be achieved naturally by a re-ordering of the arcs in the network [2,3].

The constraint matrix is also partitioned accordingly. We also require that D have full row rank, for reasons discussed in Section 3.2. Note that this does not entail any loss of generality: any SDQPP which is bounded below may be written in the above form, where Q is positive definite and D has full row rank. Section 3.1 gives a fuller discussion of this point.

In Section 2 we briefly describe the original dual-active-set method for solving the PDQPP. In Section 3 we describe conditions under which the equality subproblem generated by the SDQPP will have a unique solution, and note that these conditions imply that an initially empty active set may no longer be possible. We then show how to determine an initial active set. Step directions for the SDQPP are derived and we prove finite termination of the algorithm in a manner analogous to the proof given in [14]. From this proof, it emerges that under certain conditions, step directions cannot be determined using the method analogous to that in [14]. In Section 4 we show how step directions may be determined under these special conditions, and present the complete algorithm. We also indicate how a numerically stable implementation can be obtained as a natural extension of that given by Goldfard and Idnani.

2. The dual-active-set method of Goldfarb and Idnani

The dual-active-set method proceeds by solving quadratic equality programs. Given active set $\mathscr{A}$, the set of indices of the active constraints, the positive definite quadratic equality program PDQEP($\mathscr{A}$) is

$$\begin{array}{ll} \min & \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}Q\boldsymbol{x}+\boldsymbol{p}^{\mathrm{T}}\boldsymbol{x} \\ \text{subject to} & C_{\mathscr{A}}^{\mathrm{T}}\boldsymbol{x}=\boldsymbol{b}_{\mathscr{A}} \end{array}$$

where $C_{\mathscr{A}}$ and $\boldsymbol{b}_{\mathscr{A}}$ are just the columns and entries of C and $\boldsymbol{b}$ respectively with indices in $\mathscr{A}$. We will use this notational device throughout this section. In addition, we will use $\boldsymbol{x}_{\backslash\{j\}}$ to indicate $\boldsymbol{x}$ with the jth entry removed. For notational convenience, we will confuse a constraint in $\mathscr{A}$ with its index in $\mathscr{A}$: if constraint $i\in\mathscr{A}$ is the mth constraint in $\mathscr{A}$, then for $\boldsymbol{\lambda}\in\mathbb{R}^{|\mathscr{A}|}$ we will write λ_i instead of λ_m.

The basic form of the dual-active-set method for the PDQPP is given below, where the vector $\boldsymbol{\lambda}$ is the dual multiplier of the constraint equations.

Set $\mathscr{A}\leftarrow\emptyset$, $\boldsymbol{x}\leftarrow$ solution of the PDQEP($\mathscr{A}$), and $\boldsymbol{\lambda}\leftarrow\mathbf{0}$.
while $\exists$ a primal infeasible constraint j **do**
  Set $\lambda^{+}\leftarrow 0$.
  Find primal-dual direction $(z, \boldsymbol{r})$ and step length $t_1>0$ so that
  $(\boldsymbol{x}+t_1 z, \binom{\lambda_{\mathscr{A}}}{\lambda^{+}}+t_1\binom{-r}{1})$ solves the PDQEP($\mathscr{A}\cup\{j\}$).
  if $z=\mathbf{0}$ and $\boldsymbol{r}\leqslant\mathbf{0}$ **then** STOP (the problem is infeasible).
  while $z=\mathbf{0}$, or some dual variable would become dual infeasible,
  i.e., $\boldsymbol{\lambda}_{\mathscr{A}}-t_1\boldsymbol{r}\ngeqslant\mathbf{0}$ **do**
    Find the largest step $t_2\geqslant 0$ possible without violating dual feasibility.

Set $k \leftarrow$ index of dual variable which becomes 0.
Set $\boldsymbol{x} \leftarrow \boldsymbol{x} + t_2 \boldsymbol{z}$ and $\mathscr{A} \leftarrow \mathscr{A} \setminus \{k\}$.
Set $\lambda_k \leftarrow 0$, $\boldsymbol{\lambda}_{\mathscr{A}} \leftarrow \boldsymbol{\lambda}_{\mathscr{A}} - t_2 \boldsymbol{r}_{\setminus \{k\}}$, and $\lambda^+ \leftarrow \lambda^+ + t_2$.
Find primal-dual direction $(\boldsymbol{z}, \boldsymbol{r})$ and step length $t_1 > 0$ so that
$(\boldsymbol{x} + t_1 \boldsymbol{z}, \binom{\boldsymbol{\lambda}_{\mathscr{A}}}{\lambda^+} + t_1 \binom{-\boldsymbol{r}}{1})$ solves the PDQEP($\mathscr{A} \cup \{j\}$).
enddo
Set $\boldsymbol{x} \leftarrow \boldsymbol{x} + t_1 \boldsymbol{z}$, $\boldsymbol{\lambda}_{\mathscr{A} \cup \{j\}} \leftarrow \binom{\boldsymbol{\lambda}_{\mathscr{A}}}{\lambda^+} + t_1 \binom{-\boldsymbol{r}}{1}$, and $\mathscr{A} \leftarrow \mathscr{A} \cup \{j\}$.
enddo

The condition $\boldsymbol{z} = \boldsymbol{0}$ indicates that the addition of j to the active set $\mathscr{A}$ would result in the columns of $C_{\mathscr{A}}$ becoming linearly dependent. Throughout the algorithm, linear independence of the active constraints is maintained. Dual feasibility is also maintained, i.e. $\boldsymbol{\lambda} \geqslant \boldsymbol{0}$ at all times. The algorithm proceeds by activating primal infeasible constraints, where each activation might involve some deactivations to ensure dual feasibility, or linear independence of the active constraints. After each activation, $(\boldsymbol{x}, \boldsymbol{\lambda}_{\mathscr{A}})$ solves the PDQEP($\mathscr{A}$). Proof of finite termination follows from the fact that each activation causes a strict increase in objective function.

To find the step direction $(\boldsymbol{z}, \boldsymbol{r})$, Goldfarb and Idnani use the Moore–Penrose generalized inverse of $C_{\mathscr{A}}$ given by

$$C_{\mathscr{A}}^* = \left(C_{\mathscr{A}}^{\mathrm{T}} Q^{-1} C_{\mathscr{A}}\right)^{-1} C_{\mathscr{A}}^{\mathrm{T}} Q^{-1}$$

and the operator

$$H_{\mathscr{A}} = Q^{-1}\left(I - C_{\mathscr{A}} C_{\mathscr{A}}^*\right).$$

The initial solution of the PDQEP($\emptyset$) is given by $\boldsymbol{x} = H_{\mathscr{A}} \boldsymbol{p}$, where $H_{\mathscr{A}} = Q^{-1}$ initially. The dual direction is

$$\boldsymbol{r} = C_{\mathscr{A}}^* \boldsymbol{c}_j$$

and the primal direction is

$$\boldsymbol{z} = H_{\mathscr{A}} \boldsymbol{c}_j$$

where $\boldsymbol{c}_j$ is the jth column of C. The step length is given by

$$t_1 = \frac{b_j - \boldsymbol{c}_j^{\mathrm{T}} \boldsymbol{x}}{\boldsymbol{c}_j^{\mathrm{T}} \boldsymbol{z}}$$

provided $\boldsymbol{z} \neq \boldsymbol{0}$. When $\boldsymbol{\lambda}_{\mathscr{A}} - t_1 \boldsymbol{r} \ngeqslant \boldsymbol{0}$, the maximum step length possible without violating dual feasibility is

$$t_2 = \min_{i \in \mathscr{A},\ r_i > 0} \left\{ \frac{\lambda_i}{r_i} \right\}.$$

This minimum is achieved by some $k \in \mathscr{A}$, which is the index of the dual variable which becomes zero.

Goldfarb and Idnani do not actually compute $C_{\mathscr{A}}^*$ or $H_{\mathscr{A}}$ explicitly, but store numerically stable factorizations. Each activation or deactivation simply involves a rank one update of the factorizations.

3. Step directions for the positive semi-definite case

In this section, we show how to determine step directions in the positive semi-definite case. In doing so, we find that in addition to maintaining linear independence of the active constraints, we must maintain the rank of the submatrix of the active constraint matrix which corresponds to the "linear" variables in the partition. It is not obvious how to maintain this condition. In this section we prove that it is possible, and in Section 4.1 we describe a method of doing so.

3.1. Determining the special form of the SDQPP

We will show that it is possible to write any SDQPP in the special form we require. Consider the general SDQPP:

$$\begin{array}{ll} \min & \frac{1}{2}\tilde{\boldsymbol{x}}^{\mathrm{T}}\tilde{Q}\tilde{\boldsymbol{x}}+\tilde{\boldsymbol{p}}^{\mathrm{T}}\tilde{\boldsymbol{x}} \\ \text{subject to} & \tilde{C}^{\mathrm{T}}\tilde{\boldsymbol{x}} \geqslant \boldsymbol{b}, \end{array}$$

where $\tilde{\boldsymbol{x}}$, $\tilde{\boldsymbol{p}} \in \mathbb{R}^{q+\tilde{l}}$, $\tilde{Q}$ is a $(q+\tilde{l}) \times (q+\tilde{l})$ symmetric positive semi-definite matrix with rank q, $\tilde{C}$ is a $(q+\tilde{l}) \times c$ matrix, $\boldsymbol{b} \in \mathbb{R}^c$ and q, $\tilde{l}$, $c > 0$.

Firstly, we will show that it is possible to separate the problem into a "quadratic part" and a "linear part". This will create a problem which is in some sense "partially separable". The advantages of dealing with partially separable problems are discussed by Conn et al. in [6], and have been clearly demonstrated in a variety of contexts (see, for example, [5,7,29]).

In what follows, we will use the notation X_{nm} to denote an $n \times m$ matrix X, and $\boldsymbol{x}_{n1}$ to denote a vector $\boldsymbol{x} \in \mathbb{R}^n$. Since $\tilde{Q}$ is symmetric positive semi-definite, there exists an orthogonal matrix P such that

$$P^{\mathrm{T}}\tilde{Q}P = \begin{pmatrix} Q & 0 \\ 0 & 0 \end{pmatrix}$$

where Q is a $q \times q$ positive definite matrix. Replacing $\tilde{\boldsymbol{x}}$ by

$$\begin{pmatrix} \boldsymbol{x}_{q1} \\ \tilde{\boldsymbol{y}}_{\tilde{l}1} \end{pmatrix} \stackrel{\text{def}}{=} P^{\mathrm{T}}\tilde{\boldsymbol{x}}$$

everywhere in the SDQPP, we obtain the equivalent problem:

$$\begin{array}{ll} \min & \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}Q\boldsymbol{x}+\boldsymbol{p}^{\mathrm{T}}\boldsymbol{x}+\tilde{\boldsymbol{\rho}}^{\mathrm{T}}\tilde{\boldsymbol{y}} \\ \text{subject to} & C^{\mathrm{T}}\boldsymbol{x}+\tilde{D}^{\mathrm{T}}\tilde{\boldsymbol{y}} \geqslant \boldsymbol{b}, \end{array}$$

where

$$\begin{pmatrix} \boldsymbol{p}_{q1} \\ \tilde{\boldsymbol{\rho}}_{\tilde{l}1} \end{pmatrix} \stackrel{\text{def}}{=} P^{\mathrm{T}}\tilde{\boldsymbol{p}} \quad \text{and} \quad \begin{pmatrix} C_{qc} \\ \tilde{D}_{\tilde{l}c} \end{pmatrix} \stackrel{\text{def}}{=} P^{\mathrm{T}}\tilde{C}.$$

It only remains to show that if $\tilde{D}$ does not have full row rank, then we can obtain an equivalent problem which does. Suppose $\tilde{D}$ has row rank $l < \tilde{l}$. Without loss of generality, we may re-order the rows of $\tilde{D}$, $\tilde{\boldsymbol{\rho}}$ and $\tilde{\boldsymbol{y}}$ so that the first l rows of $\tilde{D}$ are linearly independent. Let

$$\begin{pmatrix} D_{lc} \\ E_{(\tilde{l}-l)c} \end{pmatrix} \stackrel{\text{def}}{=} \tilde{D}_{\tilde{l}c} \quad \text{and} \quad \begin{pmatrix} \boldsymbol{\rho}_{l1} \\ \boldsymbol{\sigma}_{(\tilde{l}-l)1} \end{pmatrix} \stackrel{\text{def}}{=} \tilde{\boldsymbol{\rho}}_{\tilde{l}1}.$$

Then D has full row rank and each row of E is linearly dependent on the rows of D, i.e., there exists $F_{(\tilde{l}-l)l}$ such that $E = FD$. Now provided the SDQPP is bounded below, the dual to the SDQPP has a non-empty feasible set, i.e. $\{\boldsymbol{\lambda}: \tilde{D}\boldsymbol{\lambda} = \tilde{\boldsymbol{\rho}}\} \neq \emptyset$, and so $\boldsymbol{\sigma} = F\boldsymbol{\rho}$ also. Now to transform the problem to our required form, we let

$$\begin{pmatrix} \hat{\boldsymbol{y}} \\ \hat{\boldsymbol{z}}_{(\tilde{l}-l)1} \end{pmatrix} \stackrel{\text{def}}{=} \tilde{\boldsymbol{y}}_{\tilde{l}1} \quad \text{and}\, \boldsymbol{y} \stackrel{\text{def}}{=} \hat{\boldsymbol{y}} + F^{\mathrm{T}}\hat{\boldsymbol{z}}.$$

Then $D^{\mathrm{T}}\boldsymbol{y} = \tilde{D}^{\mathrm{T}}\tilde{\boldsymbol{y}}$ and $\boldsymbol{\rho}^{\mathrm{T}}\boldsymbol{y} = \tilde{\boldsymbol{\rho}}^{\mathrm{T}}\tilde{\boldsymbol{y}}$. Consequently, the problem below is equivalent to the SDQPP:

$$\begin{array}{ll} \min & \tfrac{1}{2}\boldsymbol{x}^{\mathrm{T}}Q\boldsymbol{x} + \boldsymbol{p}^{\mathrm{T}}\boldsymbol{x} + \boldsymbol{\rho}^{\mathrm{T}}\boldsymbol{y} \\ \text{subject to} & C^{\mathrm{T}}\boldsymbol{x} + D^{\mathrm{T}}\boldsymbol{y} \geqslant \boldsymbol{b} \end{array}$$

which is exactly the form of problem we require.

As is shown by the above discussion, the transformation of the SDQPP to the special form we require presents no theoretical difficulty. However to transform an arbitrary SDQPP by numerical methods may not be straightforward. In particular, to numerically determine the rank of $\tilde{Q}$ and compute the orthogonal matrix P may be difficult. Certain classes of problem may fall naturally into the required form: network flow problems [2,3] are an important example of such a class. In general, techniques such as those discussed in [20,15,13], and more recently [8], will be needed to make the transformation practicable.

3.2. Solvability of the equality program

Consider the equality subproblem in the positive semi-definite case, the SDQEP($\mathscr{A}$):

$$\begin{array}{ll} \min & f(\boldsymbol{x}, \boldsymbol{y}) \\ \text{subject to} & C_{\mathscr{A}}^{\mathrm{T}}\boldsymbol{x} + D_{\mathscr{A}}^{\mathrm{T}}\boldsymbol{y} = \boldsymbol{b}_{\mathscr{A}}. \end{array}$$

This problem has a unique solution only if $(C_{\mathscr{A}}^{\mathrm{T}} \;\; D_{\mathscr{A}}^{\mathrm{T}})$ has linearly independent rows, and if $D_{\mathscr{A}}$ has rank equal to the dimension of the "linear" variable $\boldsymbol{y}$. We will call this dimension l. Of course, it is only possible for $D_{\mathscr{A}}$ to have rank l if it has full row rank; hence our requirement, discussed in the previous section, that D have full row rank. We will assume that the problem is not positive definite, i.e. $l > 0$. (Otherwise we may revert to the original method.)

Proposition 1. *The* SDQEP($\mathscr{A}$) *has a unique solution for every choice of* $\boldsymbol{b}_{\mathscr{A}}$, $\boldsymbol{p}$ *and* $\boldsymbol{\rho}$ *if and only if* $|\mathscr{A}| \geqslant l$, $(C_{\mathscr{A}} \;\; D_{\mathscr{A}})^{\mathrm{T}}$ *has full column rank, and the rank of* $D_{\mathscr{A}}$ *is* l.

Proof. Note firstly that if $\mathscr{A} = \emptyset$ then the SDQEP($\mathscr{A}$) either has no finite minimum, or infinitely many solutions. So $|\mathscr{A}| > 0$ is a necessary condition for a unique solution. By standard duality theory results, the SDQEP($\mathscr{A}$) has a unique solution if and only if there exists unique $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{\lambda} \in \mathbb{R}^{|\mathscr{A}|}$ such that

$$C_{\mathscr{A}}^{\mathrm{T}} \boldsymbol{x} + D_{\mathscr{A}}^{\mathrm{T}} \boldsymbol{y} = \boldsymbol{b}_{\mathscr{A}}, \tag{1}$$

$$Q\boldsymbol{x} + \boldsymbol{p} - C_{\mathscr{A}} \boldsymbol{\lambda} = \mathbf{0}, \quad \text{and} \tag{2}$$

$$\boldsymbol{\rho} - D_{\mathscr{A}} \boldsymbol{\lambda} = \mathbf{0}. \tag{3}$$

Eqs. (1) through to (3) hold if and only if

$$\begin{pmatrix} -Q & 0 & C_{\mathscr{A}} \\ 0 & 0 & D_{\mathscr{A}} \\ C_{\mathscr{A}}^{\mathrm{T}} & D_{\mathscr{A}}^{\mathrm{T}} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{y} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \boldsymbol{p} \\ \boldsymbol{\rho} \\ \boldsymbol{b}_{\mathscr{A}} \end{pmatrix}$$

where the above 3×3 block symmetric matrix is denoted by W. So the SDQEP($\mathscr{A}$) has a unique solution if and only if W is nonsingular. Now W is nonsingular implies that W has full column rank, so $(C_{\mathscr{A}}\ D_{\mathscr{A}}\ 0)^{\mathrm{T}}$ has full column rank. Hence $(C_{\mathscr{A}}\ D_{\mathscr{A}})^{\mathrm{T}}$ has full column rank. W must also have full row rank, so $(0\ 0\ D_{\mathscr{A}})$ must have full row rank, i.e. $D_{\mathscr{A}}$ must have full row rank. But $D_{\mathscr{A}}$ is an $l \times |\mathscr{A}|$ matrix, so it must be that $|\mathscr{A}| \geqslant l$ and the rank of $D_{\mathscr{A}}$ is l.

To prove the converse, we note that if $(C_{\mathscr{A}}\ D_{\mathscr{A}})^{\mathrm{T}}$ has rank $|\mathscr{A}|$ then

$$\begin{pmatrix} C_{\mathscr{A}}^{\mathrm{T}} & D_{\mathscr{A}}^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} Q^{-1} & 0 \\ 0 & I_l \end{pmatrix} \begin{pmatrix} C_{\mathscr{A}} \\ D_{\mathscr{A}} \end{pmatrix} = C_{\mathscr{A}}^{\mathrm{T}} Q^{-1} C_{\mathscr{A}} + D_{\mathscr{A}}^{\mathrm{T}} D_{\mathscr{A}}$$

also has rank $|\mathscr{A}|$ and so is invertible. Let $R_{\mathscr{A}} \stackrel{\text{def}}{=} (C_{\mathscr{A}}^{\mathrm{T}} Q^{-1} C_{\mathscr{A}} + D_{\mathscr{A}}^{\mathrm{T}} D_{\mathscr{A}})^{-1}$. Since $D_{\mathscr{A}}$ has full row rank, $D_{\mathscr{A}} R_{\mathscr{A}} D_{\mathscr{A}}^{\mathrm{T}}$ must be invertible. Let $S_{\mathscr{A}}$ be the inverse. Note that both $R_{\mathscr{A}}$ and $S_{\mathscr{A}}$ are symmetric. Now to prove the nonsingularity of W it suffices to prove the nonsingularity of

$$\Psi \stackrel{\text{def}}{=} \begin{pmatrix} C_{\mathscr{A}}^{\mathrm{T}} Q^{-1} C_{\mathscr{A}} & D_{\mathscr{A}}^{\mathrm{T}} \\ D_{\mathscr{A}} & 0 \end{pmatrix}$$

which is obtained by adding the $C_{\mathscr{A}}^{\mathrm{T}} Q^{-1}$-fold of the first row of W to the third row, dropping the first row and column, and then permuting the resulting two rows and two columns. The inverse of Ψ is explicitly given by

$$\Psi^{-1} = \begin{pmatrix} R_{\mathscr{A}} - R_{\mathscr{A}} D_{\mathscr{A}}^{\mathrm{T}} S_{\mathscr{A}} D_{\mathscr{A}} R_{\mathscr{A}} & R_{\mathscr{A}} D_{\mathscr{A}}^{\mathrm{T}} S_{\mathscr{A}} \\ S_{\mathscr{A}} D_{\mathscr{A}} R_{\mathscr{A}} & I_l - S_{\mathscr{A}} \end{pmatrix}$$

so we can deduce that W is invertible and has inverse of the form

$$W^{-1} = \begin{pmatrix} -G_{\mathscr{A}} & B_{\mathscr{A}}^{\mathrm{T}} \\ B_{\mathscr{A}} & \phi_{\mathscr{A}} \end{pmatrix}$$

where

$$G_{\mathscr{A}} = \begin{pmatrix} Q^{-1} & 0 \\ 0 & 0 \end{pmatrix} - \begin{pmatrix} C_{\mathscr{A}}^{\mathrm{T}} Q^{-1} & 0 \\ 0 & I_l \end{pmatrix}^{\mathrm{T}} \Psi^{-1} \begin{pmatrix} C_{\mathscr{A}}^{\mathrm{T}} Q^{-1} & 0 \\ 0 & I_l \end{pmatrix},$$

$$B_{\mathscr{A}} = \begin{pmatrix} (\Psi^{-1})_{1,1} & (\Psi^{-1})_{1,2} \end{pmatrix} \begin{pmatrix} C_{\mathscr{A}}^{\mathrm{T}} Q^{-1} & 0 \\ 0 & I_l \end{pmatrix}, \quad \text{and} \quad \phi_{\mathscr{A}} = (\Psi^{-1})_{1,1}. \qquad \square.$$

In Section 3.4 we discuss $G_{\mathscr{A}}$ and $B_{\mathscr{A}}$ further, since these matrices play an important role in the algorithm we develop. The matrix $(\Psi^{-1})_{1,1}$ is also used directly in the algorithm, as will be seen in Section 4.2.

3.3. The initial active set

The first point that emerges from Proposition 1 is that the initial active set can no longer be empty. It is necessary to start with an initial active set which will meet the conditions of Proposition 1 and which results in a feasible initial dual variable.

To satisfy the minimal requirements of the initial active set, we need to find $\mathscr{A}$ with $|\mathscr{A}| = l$ and $D_{\mathscr{A}}$ nonsingular. The latter condition implies that $(C_{\mathscr{A}}^{\mathrm{T}}\ D_{\mathscr{A}}^{\mathrm{T}})$ will have linearly independent rows. Now, from Eq. (3), the dual solution to the SQDEP($\mathscr{A}$) will be given by $D_{\mathscr{A}}^{-1}\boldsymbol{\rho}$, so the problem of finding an initial active set becomes the problem:

Find $\mathscr{A}$ so that
(i) $|\mathscr{A}| = l$,
(ii) $D_{\mathscr{A}}$ is invertible, and
(iii) $D_{\mathscr{A}}^{-1}\boldsymbol{\rho} \geqslant \mathbf{0}$.

This is equivalent to the problem solved in the first phase of the simplex method for finding an initial basic feasible solution of a linear programming problem. In this case, the linear programming problem would be a problem in $\boldsymbol{\lambda}$ with feasible region given by:

$$D\boldsymbol{\lambda} = \boldsymbol{\rho}, \quad \text{and} \quad \boldsymbol{\lambda} \geqslant 0.$$

This may be solved by any of the usual methods employed for the first phase of the simplex method. If this feasible region is empty, then since it is actually the feasible region for the dual to the SDQPP, the SDQPP must be unbounded.

3.4. Matrices for step directions

In this section, we discuss the matrices $B_{\mathscr{A}}$ and $G_{\mathscr{A}}$, which are analogous to the matrices $C_{\mathscr{A}}^{*}$ and $H_{\mathscr{A}}$ of Section 2. They can be expressed explicitly using the matrices $R_{\mathscr{A}}$ and $S_{\mathscr{A}}$ from Proposition 1:

$$B_{\mathscr{A}} \stackrel{\text{def}}{=} \left(\left(I_{|\mathscr{A}|} - R_{\mathscr{A}} D_{\mathscr{A}}^{\mathrm{T}} S_{\mathscr{A}} D_{\mathscr{A}} \right) R_{\mathscr{A}} C_{\mathscr{A}}^{\mathrm{T}} Q^{-1} \quad R_{\mathscr{A}} D_{\mathscr{A}}^{\mathrm{T}} S_{\mathscr{A}} \right)$$

and

$$G_{\mathscr{A}} \stackrel{\text{def}}{=} \begin{pmatrix} Q^{-1}\left(I_{|\mathscr{A}|} - C_{\mathscr{A}} R_{\mathscr{A}} \left(I_{|\mathscr{A}|} - D_{\mathscr{A}}^{\mathrm{T}} S_{\mathscr{A}} D_{\mathscr{A}} R_{\mathscr{A}} \right) C_{\mathscr{A}}^{\mathrm{T}} Q^{-1} \right) & -Q^{-1} C_{\mathscr{A}} R_{\mathscr{A}} D_{\mathscr{A}}^{\mathrm{T}} S_{\mathscr{A}} \\ -S_{\mathscr{A}} D_{\mathscr{A}} R_{\mathscr{A}} C_{\mathscr{A}}^{\mathrm{T}} Q^{-1} & S_{\mathscr{A}} - I_L \end{pmatrix}.$$

Recall that both $R_{\mathscr{A}}$ and $S_{\mathscr{A}}$ are symmetric. Clearly $G_{\mathscr{A}}$ is also a symmetric matrix. Note that if the problem is actually positive definite, so that $l = 0$, $D_{\mathscr{A}}$ and $S_{\mathscr{A}}$ are not defined, and $R_{\mathscr{A}} = (C_{\mathscr{A}}^{\mathrm{T}} Q^{-1} C_{\mathscr{A}})^{-1}$, then $B_{\mathscr{A}}$ and $G_{\mathscr{A}}$ do revert to the original $C_{\mathscr{A}}^{*}$ and $H_{\mathscr{A}}$.

In what follows, we will, to a large extent, be imitating the results given in [14] by Goldfarb and Idnani for the positive definite case. Here we will be showing why these results hold for the new matrices defined above. We claim that, with the modifications discussed in Section 4.1 and with the initial active set found as in Section 3.3, the SDQPP may be solved by the algorithm given in Section 2, with the matrices $C^*_{\mathscr{A}}$ and $H_{\mathscr{A}}$ replaced everywhere by $B_{\mathscr{A}}$ and $G_{\mathscr{A}}$ respectively.

For notational convenience, we will from this point onwards be omitting the subscripts $\mathscr{A}$. To avoid confusion, we will let N be $C_{\mathscr{A}}$ and M be $D_{\mathscr{A}}$. We will also omit the subscript indicating the dimension of an identity matrix. In general, this dimension can be inferred from the context.

The following properties of the matrices B and G will be of use:

$$G \text{ is positive semi-definite,} \tag{4}$$

$$(\boldsymbol{n}^{\mathrm{T}}\ \boldsymbol{m}^{\mathrm{T}})G\begin{pmatrix}\boldsymbol{n}\\ \boldsymbol{m}\end{pmatrix}=0 \quad\Rightarrow\quad \begin{pmatrix}\boldsymbol{n}\\ \boldsymbol{m}\end{pmatrix} \text{ is in the column space of } \begin{pmatrix}N\\ M\end{pmatrix}, \tag{5}$$

$$G\begin{pmatrix}N\\ M\end{pmatrix}=0, \tag{6}$$

$$B\begin{pmatrix}N\\ M\end{pmatrix}=I, \tag{7}$$

$$\begin{pmatrix}Q & 0\\ 0 & 0\end{pmatrix}G=I-\begin{pmatrix}N\\ M\end{pmatrix}B, \tag{8}$$

and

$$G\begin{pmatrix}Q & 0\\ 0 & 0\end{pmatrix}G=G. \tag{9}$$

The properties (6), (7) and (8) all follow directly from the identity

$$W^{-1}W=\begin{pmatrix}I_{q+1} & 0\\ 0 & I_{|\mathscr{A}|}\end{pmatrix}$$

using W as defined in Proposition 1, and property (9) follows easily from (6) and (8). Property (4) follows from (9) and the positive definiteness of Q. To prove property (5) holds requires somewhat more effort. We firstly observe that if

$$(\boldsymbol{n}^{\mathrm{T}}\ \boldsymbol{m}^{\mathrm{T}})G\begin{pmatrix}\boldsymbol{n}\\ \boldsymbol{m}\end{pmatrix}=0$$

then

$$(\boldsymbol{n}^{\mathrm{T}}\ \boldsymbol{m}^{\mathrm{T}}\ \boldsymbol{0}^{\mathrm{T}})W^{-1}\begin{pmatrix}\boldsymbol{n}\\ \boldsymbol{m}\\ \boldsymbol{0}\end{pmatrix}=0$$

and so

$$(\boldsymbol{\alpha}^{\mathrm{T}}\ \boldsymbol{\beta}^{\mathrm{T}}\ \boldsymbol{\gamma}^{\mathrm{T}})W\begin{pmatrix}\boldsymbol{\alpha}\\ \boldsymbol{\beta}\\ \boldsymbol{\gamma}\end{pmatrix}=0 \tag{10}$$

where $(\boldsymbol{\alpha}\ \boldsymbol{\beta}\ \boldsymbol{\gamma})^{\mathrm{T}}$ is the unique solution of

$$W\begin{pmatrix}\boldsymbol{\alpha}\\ \boldsymbol{\beta}\\ \boldsymbol{\gamma}\end{pmatrix}=\begin{pmatrix}\boldsymbol{n}\\ \boldsymbol{m}\\ \boldsymbol{0}\end{pmatrix}, \tag{11}$$

which is equivalent to

$$\begin{pmatrix}-Q\boldsymbol{\alpha}+N\boldsymbol{\gamma}\\ M\boldsymbol{\gamma}\\ N^{\mathrm{T}}\boldsymbol{\alpha}+M^{\mathrm{T}}\boldsymbol{\beta}\end{pmatrix}=\begin{pmatrix}\boldsymbol{n}\\ \boldsymbol{m}\\ \boldsymbol{0}\end{pmatrix}. \tag{12}$$

From (10) we can deduce that

$$-\boldsymbol{\alpha}^{\mathrm{T}}Q\boldsymbol{\alpha}+2\boldsymbol{\gamma}^{\mathrm{T}}(N^{\mathrm{T}}\boldsymbol{\alpha}+M^{\mathrm{T}}\boldsymbol{\beta})=0$$

and so $\boldsymbol{\alpha}^{\mathrm{T}}Q\boldsymbol{\alpha}=0$ since from (11), using the last row of W, it must be that $N^{\mathrm{T}}\boldsymbol{\alpha}+M^{\mathrm{T}}\boldsymbol{\beta}=0$. Consequently $\boldsymbol{\alpha}=\boldsymbol{0}$ since Q is positive definite. Thus from (12), we have that

$$\begin{pmatrix}\boldsymbol{n}\\ \boldsymbol{m}\end{pmatrix}=\begin{pmatrix}N\boldsymbol{\gamma}\\ M\boldsymbol{\gamma}\end{pmatrix},$$

i.e. $(\boldsymbol{n}\ \boldsymbol{m})^{\mathrm{T}}$ is in the column space of $(N\ M)^{\mathrm{T}}$.

The matrices B and G encapsulate the optimality conditions of the SDQEP($\mathscr{A}$) as shown in the following proposition.

Proposition 2. *Primal variables* $(\boldsymbol{x}, \boldsymbol{y})$ *and dual variable* $\boldsymbol{\lambda}$ *satisfy Eqs.* (2) *and* (3) *if and only if*

$$G\nabla f(\boldsymbol{x}, \boldsymbol{y})=\boldsymbol{0} \tag{13}$$

and

$$\boldsymbol{\lambda}=B\nabla f(\boldsymbol{x}, \boldsymbol{y}). \tag{14}$$

Proof. $(\Rightarrow)$ From (2) and (3) we have that

$$\begin{pmatrix}N\\ M\end{pmatrix}\boldsymbol{\lambda}=\nabla f(\boldsymbol{x}, \boldsymbol{y}) \tag{15}$$

and so

$$\boldsymbol{\lambda}=B\nabla f(\boldsymbol{x}, \boldsymbol{y}) \tag{16}$$

by (7). Also

$$\begin{aligned}G\nabla f(\boldsymbol{x}, \boldsymbol{y}) &= G\begin{pmatrix}N\\ M\end{pmatrix}\boldsymbol{\lambda} \qquad \text{from (15)}\\ &= \boldsymbol{0} \qquad \text{by (6)}.\end{aligned}$$

$(\Leftarrow)$ Conversely,

$$\begin{aligned}\begin{pmatrix}N\\ M\end{pmatrix}\boldsymbol{\lambda} &= \begin{pmatrix}N\\ M\end{pmatrix}B\nabla f(\boldsymbol{x}, \boldsymbol{y}) \qquad \text{from (14)}\\ &= -\left\{I-\begin{pmatrix}N\\ M\end{pmatrix}B\right\}\nabla f(\boldsymbol{x}, \boldsymbol{y})+\nabla f(\boldsymbol{x}, \boldsymbol{y})\end{aligned}$$

$$= -\begin{pmatrix} Q & 0 \\ 0 & 0 \end{pmatrix} G\nabla f(x, y) + \nabla f(x, y) \qquad \text{by (8)}$$

$$= \nabla f(x, y) \qquad \text{from (13)}. \qquad \square$$

We will now prove finite termination of the dual-active-set method for the SDQPP. This is just the method given in Section 2 with matrices $C^*_{\mathscr{A}}$ and $H_{\mathscr{A}}$ replaced by B and G, and with the initial active set found as in Section 3.3. The original method is also modified to handle the possibility that while some constraint is being activated, a deactivation causes M to lose rank. M must have full row rank in order to compute the step directions in the way described in Section 2. However step directions can still be found, as is shown in the proofs below. A method of finding them is described in Section 4.1.

The following definition is used throughout the proof.

Definition 1. A triple $((x, y), \mathscr{A}, j)$ is said to be a *V(violated)-triple* if $M^+ \stackrel{\text{def}}{=} D_{\mathscr{A} \cup \{j\}}$ has rank l, $(N^+ \ M^+)^{\mathrm{T}}$ has full column rank, where $N^+ \stackrel{\text{def}}{=} C_{\mathscr{A} \cup \{j\}}$,

$$s_j(x, y) < 0$$

where $s_i(x, y) \stackrel{\text{def}}{=} c_i^{\mathrm{T}} x + d_i^{\mathrm{T}} y - b_i$ is the slack in the ith constraint,

$$s_i(x, y) = 0, \quad \forall i \in \mathscr{A}, \tag{17}$$

$$G^+ \nabla f(x, y) = 0, \quad \text{and} \tag{18}$$

$$\lambda^+ \stackrel{\text{def}}{=} B^+ \nabla f(x, y) \geqslant 0, \tag{19}$$

where $G^+ \stackrel{\text{def}}{=} G_{\mathscr{A} \cup \{j\}}$ and $B^+ \stackrel{\text{def}}{=} B_{\mathscr{A} \cup \{j\}}$.

In the dual-active-set method, throughout the activation of some constraint j, the triple $((x, y), \mathscr{A}, j)$ is always a V-triple. The following lemma shows that if we start with a V-triple, and move a short distance along the primal-dual step direction $((z, w), r)$ defined by the B and G matrices, then the result is again a V-triple. It also shows that if we take a step of length

$$t_1 \stackrel{\text{def}}{=} -\frac{s_j(x, y)}{c_j^{\mathrm{T}} z + d_j^{\mathrm{T}} w},$$

then we reach a solution of the SDQEP($\mathscr{A} \cup \{j\}$).

Lemma 1. *Let $((x, y), \mathscr{A}, j)$ be a V-triple and consider points of the form*

$$\begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + s\begin{pmatrix} z \\ w \end{pmatrix} \tag{20}$$

where $s > 0$ and

$$\begin{pmatrix} z \\ w \end{pmatrix} = G\begin{pmatrix} c_j \\ d_j \end{pmatrix}. \tag{21}$$

Then

$$G^+ \nabla f(\bar{x}, \bar{y}) = \mathbf{0}, \tag{22}$$

$$s_i(\bar{x}, \bar{y}) = 0, \quad \forall i \in \mathscr{A}, \quad \textit{and} \tag{23}$$

$$B^+ \nabla f(\bar{x}, \bar{y}) = \boldsymbol{\lambda}^+ + s\begin{pmatrix} -\boldsymbol{r} \\ 1 \end{pmatrix} \tag{24}$$

where

$$\boldsymbol{r} = B\begin{pmatrix} \boldsymbol{c}_j \\ \boldsymbol{d}_j \end{pmatrix}, \tag{25}$$

and

$$s_j(\bar{x}, \bar{y}) = s_j(x, y) + s(z^{\mathrm{T}} \;\; w^{\mathrm{T}})\begin{pmatrix} \boldsymbol{c}_j \\ \boldsymbol{d}_j \end{pmatrix}. \tag{26}$$

Proof. The following can be proved directly from definitions:

$$\nabla f(\bar{x}, \bar{y}) = \nabla f(x, y) + s\begin{pmatrix} Qz \\ \mathbf{0} \end{pmatrix}, \tag{27}$$

and

$$\begin{pmatrix} Qz \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} N^+ \\ M^+ \end{pmatrix}\begin{pmatrix} -\boldsymbol{r} \\ 1 \end{pmatrix}. \tag{28}$$

Now

$$\begin{aligned} G^+ \nabla f(\bar{x}, \bar{y}) &= G^+ \nabla f(x, y) + sG^+\begin{pmatrix} Qz \\ \mathbf{0} \end{pmatrix} && \text{by (27)} \\ &= sG^+\begin{pmatrix} N^+ \\ M^+ \end{pmatrix}\begin{pmatrix} -\boldsymbol{r} \\ 1 \end{pmatrix} && \text{by (18) and (28)} \\ &= \mathbf{0} && \text{by (6),} \end{aligned}$$

and for each $i \in \mathscr{A}$,

$$\begin{aligned} s_i(\bar{x}, \bar{y}) &= s_i(x, y) + s\left(\boldsymbol{c}_i^{\mathrm{T}} \;\; \boldsymbol{d}_i^{\mathrm{T}}\right)\begin{pmatrix} z \\ w \end{pmatrix} \\ &= s\left(\boldsymbol{c}_i^{\mathrm{T}} \;\; \boldsymbol{d}_i^{\mathrm{T}}\right)G\begin{pmatrix} \boldsymbol{c}_j \\ \boldsymbol{d}_j \end{pmatrix} && \text{by (17) and (21)} \\ &= 0 && \text{by (6).} \end{aligned}$$

In addition,

$$\begin{aligned} B^+ \nabla f(\bar{x}, \bar{y}) &= B^+ \nabla f(x, y) + sB^+\begin{pmatrix} Qz \\ \mathbf{0} \end{pmatrix} && \text{by (27)} \\ &= \boldsymbol{\lambda}^+ + sB^+\begin{pmatrix} N^+ \\ M^+ \end{pmatrix}\begin{pmatrix} -\boldsymbol{r} \\ 1 \end{pmatrix} && \text{by (19) and (28)} \\ &= \boldsymbol{\lambda}^+ + s\begin{pmatrix} -\boldsymbol{r} \\ 1 \end{pmatrix} && \text{by (7).} \end{aligned} \tag{29}$$

Eq. (26) follows straight from the definitions. □

It is obvious from (4), (5) and the definition of a V-triple that

$$(z^{\mathrm{T}}\ w^{\mathrm{T}})\begin{pmatrix} c_j \\ d_j \end{pmatrix} = \begin{pmatrix} c_j^{\mathrm{T}} & d_j^{\mathrm{T}} \end{pmatrix} G \begin{pmatrix} c_j \\ d_j \end{pmatrix} > 0 \tag{30}$$

and so from (26) in the above lemma we can see that $s < t_1$ if and only if $s_j(\bar{x}, \bar{y}) < 0$ and $s = t_1$ if and only if $s_j(\bar{x}, \bar{y}) = 0$. Now by Proposition 2, we have that if $s = t_1$ and $B^+ \nabla f(\bar{x}, \bar{y}) \geqslant 0$ then $(\bar{x}, \bar{y})$ solves the SDQEP($\mathscr{A} \cup \{j\}$). Otherwise, there is some largest $s = t_2 < t_1$ with $B^+ \nabla f(\bar{x}, \bar{y}) \geqslant 0$ and with some component of $\boldsymbol{\lambda}^+ - t_2 \boldsymbol{r}$ decreased to zero. If $k \in \mathscr{A}$ is the constraint corresponding to this component, and $(\bar{x}, \bar{y})$ is given by (20) with $s = t_2$, then $((\bar{x}, \bar{y}), \mathscr{A} \setminus \{k\}, j)$ is a V-triple. This is proved in the following theorem. However we first prove that $M^+_{\mathscr{A} \setminus \{k\}}$ must have full row rank, which is a necessary condition for $((\bar{x}, \bar{y}), \mathscr{A} \setminus \{k\}, j)$ to be a V-triple.

Lemma 2. *Let $((x, y), \mathscr{A}, j)$ be a V-triple, $\boldsymbol{r}$ be defined by* (25), *and suppose that $\boldsymbol{\lambda}^+ + t_1\binom{-r}{1} \ngeqslant \mathbf{0}$, where $\boldsymbol{\lambda}^+$ is defined by* (19). *Let t_2 and the constraint k be chosen to satisfy:*

$$t_2 \overset{\text{def}}{=} \min_{i \in \mathscr{A},\, r_i > 0} \left\{ \frac{\lambda_i^+}{r_i} \right\} = \frac{\lambda_k^+}{r_k}. \tag{31}$$

Then $M^+_{\mathscr{A} \setminus \{k\}}$ has full row rank.

Proof. We have that

$$M^+ \begin{pmatrix} -r \\ 1 \end{pmatrix} = \mathbf{0} \qquad \text{from (28)}$$

and $r_k > 0$ from (31), so

$$\boldsymbol{d}_k = \frac{1}{r_k}\left\{ \boldsymbol{d}_j - \sum_{i \in \mathscr{A} \setminus \{k\}} r_i \boldsymbol{d}_i \right\}.$$

Hence the column space of M is contained in the column space of $M^+_{\mathscr{A} \setminus \{k\}}$, so rank $(M) \leqslant \operatorname{rank}(M^+_{\mathscr{A} \setminus \{k\}})$. But rank $(M) = l$, so rank $(M^+_{\mathscr{A} \setminus \{k\}}) = l$ and thus $M^+_{\mathscr{A} \setminus \{k\}}$ has full row rank. □

Theorem 1. *Given $((x, y), \mathscr{A}, j)$ a V-triple and $\boldsymbol{r}$ defined by* (25), *if $\boldsymbol{\lambda}^+ - t_1 \boldsymbol{r} \geqslant 0$ set $s = t_1$; otherwise set $s = t_2$ and define k according to* (31). *Define $(\bar{x}, \bar{y})$ by* (20) *and* (21). *Then*

$$s_j(\bar{x}, \bar{y}) \geqslant s_j(x, y) \tag{32}$$

and

$$f(\bar{x}, \bar{y}) \geqslant f(x, y), \tag{33}$$

with $f(\bar{x}, \bar{y}) = f(x, y)$ if and only if $s = t_2 = 0$, or $\boldsymbol{\lambda}_j^+ = 0$ and $z = \mathbf{0}$. Furthermore, if $s = t_2$ then $((\bar{x}, \bar{y}), \mathscr{A} \setminus \{k\}, j)$ is a V-triple. In addition, if $s = t_1$ then $(\bar{x}, \bar{y})$ solves the SDQEP($\mathscr{A} \cup \{j\}$).

Proof. The inequality (32) follows directly from Lemma 1 via (26) and (30). Now from the definition of $(\bar{x}, \bar{y})$,

$$f(\bar{x}, \bar{y}) - f(x, y) = s(z^{\mathrm{T}} \ \ w^{\mathrm{T}})\nabla f(x, y) + \tfrac{1}{2}s^2 z^{\mathrm{T}} Qz.$$

Since $((x, y), \mathscr{A}\setminus\{k\}, j)$ is a V-triple, $G^+ \nabla f(x, y) = \mathbf{0}$ and hence

$$\begin{aligned}\nabla f(x, y) &= \begin{pmatrix} N^+ \\ M^+ \end{pmatrix} B^+ \nabla f(x, y) \\ &= \begin{pmatrix} N^+ \\ M^+ \end{pmatrix} \boldsymbol{\lambda}^+ = \begin{pmatrix} N & c_j \\ M & d_j \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda}^+_{\mathscr{A}} \\ \lambda^+_j \end{pmatrix}.\end{aligned}$$

Consequently

$$\begin{aligned}G\nabla f(x, y) &= G\left\{\begin{pmatrix} N \\ M \end{pmatrix} \boldsymbol{\lambda}^+_{\mathscr{A}} + \begin{pmatrix} c_j \\ d_j \end{pmatrix} \lambda^+_j\right\} \\ &= G\begin{pmatrix} c_j \\ d_j \end{pmatrix} \lambda^+_j \qquad \text{from (6)}\end{aligned}$$

and hence

$$(z^{\mathrm{T}} \ \ w^{\mathrm{T}})\nabla f(x, y) = \lambda^+_j \left(c_j^{\mathrm{T}} \ \ d_j^{\mathrm{T}}\right) G \begin{pmatrix} c_j \\ d_j \end{pmatrix}.$$

So

$$f(\bar{x}, \bar{y}) - f(x, y) = s\lambda^+_j \left(c_j^{\mathrm{T}} \ \ d_j^{\mathrm{T}}\right) G \begin{pmatrix} c_j \\ d_j \end{pmatrix} + \tfrac{1}{2}s^2 z^{\mathrm{T}} Qz \geqslant 0 \tag{34}$$

by (5) and the definition of a V-triple, and since $s \geqslant 0$ and $\lambda^+_j \geqslant 0$ by (19). Now $t_1 > 0$, since from $((x, y), \mathscr{A}, j)$ a V-triple we have that $s_j(x, y) < 0$. Also

$$\left(c_j^{\mathrm{T}} \ \ d_j^{\mathrm{T}}\right) G \begin{pmatrix} c_j \\ d_j \end{pmatrix} > 0$$

by (4), (5) and the definition of a V-triple, so $f(\bar{x}, \bar{y}) = f(x, y)$ if and only if $s = t_2 = 0$, or $\lambda^+_j = 0$ and $z = \mathbf{0}$. We will now prove that if $s = t_2$ then $((\bar{x}, \bar{y}), \mathscr{A}\setminus\{k\}, j)$ is a V-triple. Firstly, is is clear from (30) and the subsequent discussion that $s_j(\bar{x}, \bar{y}) < 0$. It is also obvious from (23) that $s_i(\bar{x}, \bar{y}) = 0$ for all $i \in \mathscr{A}\setminus\{k\}$. Note that by Lemma 2, $M^+_{\mathscr{A}\setminus\{k\}}$ has full row rank, so $G^+_{\mathscr{A}\setminus\{k\}}$ and $B^+_{\mathscr{A}\setminus\{k\}}$ can be defined. From (22) and (24) we have that

$$\nabla f(\bar{x}, \bar{y}) = \begin{pmatrix} N^+ \\ M^+ \end{pmatrix} \left\{\boldsymbol{\lambda}^+ + t_2 \begin{pmatrix} -\boldsymbol{r} \\ 1 \end{pmatrix}\right\}$$

so

$$\begin{aligned}G^+_{\mathscr{A}\setminus\{k\}}\nabla f(\bar{x}, \bar{y}) &= (\lambda^+_k - t_2 r_k) G^+_{\mathscr{A}\setminus\{k\}} \begin{pmatrix} c_k \\ d_k \end{pmatrix} \qquad \text{by (6)} \\ &= \mathbf{0}\end{aligned}$$

by the definition of t_2. Also

$$B^+_{\mathscr{A}\setminus\{k\}}\nabla f(\bar{x},\bar{y}) = \boldsymbol{\lambda}^+_{\setminus\{k\}} + t_2\begin{pmatrix} -\boldsymbol{r}_{\setminus\{k\}} \\ 1 \end{pmatrix} + (\lambda^+_k - t_2 r_k) B^+_{\mathscr{A}\setminus\{k\}}\begin{pmatrix} \boldsymbol{c}_k \\ \boldsymbol{d}_k \end{pmatrix} \quad \text{by (7)}$$

$$= (B^+\nabla f(\bar{x},\bar{y}))_{\setminus\{k\}} \quad \text{by (24) and the definition of } k$$

$$\geqslant \mathbf{0}$$

since t_2 is chosen so that $B^+\nabla f(\bar{x},\bar{y}) \geqslant \mathbf{0}$. This completes the proof that $((\bar{x},\bar{y}), \mathscr{A}\setminus\{k\}, j)$ is a V-triple. Now if $s = t_1$ then $B^+\nabla f(\bar{x},\bar{y}) \geqslant \mathbf{0}$ by Lemma 1 and the definition of t_1. Setting the new dual variable $\boldsymbol{\lambda}_{\mathscr{A}\cup\{j\}} = B^+\nabla f(\bar{x},\bar{y})$, we thus have dual feasibility. In addition, $(\bar{x},\bar{y})$ and $\boldsymbol{\lambda}_{\mathscr{A}\cup\{j\}}$ satisfy the optimality conditions for the SDQEP($\mathscr{A}\cup\{j\}$) by (22) and Proposition 2. We also have primal feasibility by (23) and since $s_j(\bar{x},\bar{y}) = 0$ by (26) and the definition of t_1. Hence $(\bar{x},\bar{y})$ and $\boldsymbol{\lambda}_{\mathscr{A}\cup\{j\}}$ solve the SDQEP($\mathscr{A}\cup\{j\}$). □

In order to prove finite termination of the algorithm, we require that the objective function be non-decreasing at each step, and be constant for at most a finite number of steps in succession. Theorem 1 shows that if we start from a V-triple, then our step will not decrease the objective function. However, if $s = t_2 = 0$, or $\lambda^+_j = 0$ and $z = \mathbf{0}$, then the objective function will be constant (otherwise it will increase). These two conditions under which the objective function will be constant can occur only a finite number of times in succession since from (29) we see that λ^+_j can only be zero as long as $s = 0$, which can only occur if $s = t_2 = 0$, but each time $s = t_2 = 0$ we can remove a constraint from the active set to obtain a new V-triple; clearly this can be done at most a finite number of times. Thus Theorem 1 alone would be sufficient to furnish a proof of finite termination of the algorithm, provided that at the start of each activation $((x, y), \mathscr{A}, j)$ is a V-triple. If $((x, y), \mathscr{A}, j)$ is not a V-triple at the start of some activation, then since (x, y) solves the SDQEP($\mathscr{A}$), it must be that $(N^+\ M^+)^{\mathrm{T}}$ does not have full column rank, i.e. $(\boldsymbol{c}_j\ \boldsymbol{d}_j)^{\mathrm{T}}$ is in the column space of $(N\ M)^{\mathrm{T}}$. In this case, $(z\ w)^{\mathrm{T}} = \mathbf{0}$ by (6). If $\boldsymbol{r} \leqslant 0$ then $\boldsymbol{\lambda}_{\mathscr{A}} - s\boldsymbol{r} \geqslant \mathbf{0}$ for all $s > 0$, so the dual problem is unbounded and hence the primal problem is infeasible. Otherwise, some constraint k will be dropped from the active set so that $((x, y), \mathscr{A}\setminus\{k\}, j)$ is a V-triple. This is proved in the theorem below. In this case there is no change in the objective function, since the deactivation does not change the primal variables.

Theorem 2. *Let (x, y) and $\boldsymbol{\lambda}$ solve the* SDQEP($\mathscr{A}$) *and let $j \notin \mathscr{A}$ be a constraint with $s_j(x, y) < 0$ and $(\boldsymbol{c}_j\ \boldsymbol{d}_j)^{\mathrm{T}}$ in the column space of $(N\ M)^{\mathrm{T}}$. If some component of $\boldsymbol{r}$ defined by* (25) *is positive, we can drop constraint k from the active set, where k is determined by*

$$\frac{\lambda_k}{r_k} = \min_{i\in\mathscr{A},\, r_i>0}\left\{\frac{\lambda_i}{r_i}\right\}, \tag{35}$$

to give $((x, y), \mathscr{A}\setminus\{k\}, j)$ a V-triple. Otherwise we must declare the problem to be infeasible.

Proof. Suppose that $r \leqslant 0$ and the SDQEP($\mathscr{A} \cup \{j\}$) is feasible. Then there exists some feasible solution

$$\begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} z \\ w \end{pmatrix}$$

of the SDQEP($\mathscr{A} \cup \{j\}$). So $s_j(\bar{x}, \bar{y}) = 0$ and hence

$$\begin{pmatrix} c_j^{\mathrm{T}} & d_j^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} z \\ w \end{pmatrix} > 0.$$

Now since $(c_j\ d_j)^{\mathrm{T}}$ is in the column space of $(N\ M)^{\mathrm{T}}$, it must be that

$$\begin{pmatrix} c_j \\ d_j \end{pmatrix} = \begin{pmatrix} N \\ M \end{pmatrix} r \tag{36}$$

by (25) and (7), so

$$r^{\mathrm{T}} (N^{\mathrm{T}}\ M^{\mathrm{T}}) \begin{pmatrix} z \\ w \end{pmatrix} > 0. \tag{37}$$

But $s_i(\bar{x}, \bar{y}) = 0$ and $s_i(x, y) = 0$ for all $i \in \mathscr{A}$ so

$$(N^{\mathrm{T}}\ M^{\mathrm{T}}) \begin{pmatrix} z \\ w \end{pmatrix} = \mathbf{0}$$

which contradicts (37). Thus the SDQEP($\mathscr{A} \cup \{j\}$) must be infeasible.

If $r \nleqslant 0$ then we have k defined by (35) with $r_k > 0$. Now $(N^+_{\mathscr{A}\setminus\{k\}}\ M^+_{\mathscr{A}\setminus\{k\}})^{\mathrm{T}}$ has full column rank from (36) and since $(N\ M)^{\mathrm{T}}$ has full column rank. From (36) we have that

$$\begin{pmatrix} c_k \\ d_k \end{pmatrix} = \frac{1}{r_k} \left\{ \begin{pmatrix} c_j \\ d_j \end{pmatrix} - \sum_{i \in \mathscr{A}\setminus\{k\}} r_i \begin{pmatrix} c_i \\ d_i \end{pmatrix} \right\} \tag{38}$$

which can be used to show that the column space of M is contained in the column space of $M^+_{\mathscr{A}\setminus\{k\}}$. Since M has full row rank, it must be that $M^+_{\mathscr{A}\setminus\{k\}}$ does too. We have that $s_i(x, y) = 0$ for all $i \in \mathscr{A}\setminus\{k\}$ since (x, y) solves the SDQEP($\mathscr{A}$). From Proposition 2, we also have that

$$\begin{aligned} \nabla f(x, y) &= \begin{pmatrix} N \\ M \end{pmatrix} \boldsymbol{\lambda} \\ &= \sum_{i \in \mathscr{A}\setminus\{k\}} \lambda_i \begin{pmatrix} c_i \\ d_i \end{pmatrix} + \frac{\lambda_k}{r_k} \left\{ \begin{pmatrix} c_j \\ d_j \end{pmatrix} - \sum_{i \in \mathscr{A}\setminus\{k\}} r_i \begin{pmatrix} c_i \\ d_i \end{pmatrix} \right\} \qquad \text{by (38)} \\ &= \begin{pmatrix} N^+_{\mathscr{A}\setminus\{k\}} \\ M^+_{\mathscr{A}\setminus\{k\}} \end{pmatrix} \left\{ \begin{pmatrix} \boldsymbol{\lambda}_{\setminus\{k\}} \\ 0 \end{pmatrix} + \frac{\lambda_k}{r_k} \begin{pmatrix} -r \\ 1 \end{pmatrix} \right\}, \end{aligned}$$

and so

$$B^+_{\mathscr{A}\setminus\{k\}} \nabla f(x, y) = \begin{pmatrix} \boldsymbol{\lambda}_{\setminus\{k\}} \\ 0 \end{pmatrix} + \frac{\lambda_k}{r_k} \begin{pmatrix} -r \\ 1 \end{pmatrix} \geqslant 0$$

by (7) and (35). Also $G^+_{\mathscr{A}\setminus\{k\}} \nabla f(x, y) = \mathbf{0}$ by (6). Hence $((x, y), \mathscr{A}\setminus\{k\}, j)$ is a V-triple. □

If (x, y) solves the SDQEP($\mathscr{A}$) before some activation, say constraint j is to be activated, and if $(\bar{x}, \bar{y})$ is the solution after activation, so $(\bar{x}, \bar{y})$ solves the SDQEP($\overline{\mathscr{A}} \cup \{j\}$) where $\overline{\mathscr{A}} \subseteq \mathscr{A}$, then $f(\bar{x}, \bar{y}) > f(x, y)$ by Theorems 1 and 2. Furthermore, such an $(\bar{x}, \bar{y})$ will be reached in a finite number of steps, since each time $\boldsymbol{\lambda}^+ - t_1 \boldsymbol{r} \not\geq \mathbf{0}$, s is set to t_2 and a constraint is dropped from the active set; the active set is finite so this can only occur a finite number of times. So an active set can never re-occur, and since there is only a finite number of possible active sets, the following theorem must hold.

Theorem 3. *The algorithm will solve the* SDQPP, *or indicate that it has no feasible solution, in a finite number of steps.*

4. The dual-active-set method for positive semi-definite quadratic programming

In Lemma 2 we showed that during an activation it was impossible for M^+ to lose rank. However in the original Goldfarb and Idnani method, the step direction would be determined using B and G, not B^+ or G^+. Unfortunately, we cannot guarantee that M will not lose rank after the deactivation of some constraint. If this occurs we cannot calculate B or G and hence cannot obtain step directions using these matrices. Instead we "look ahead" by calculating the primal-dual solution of the SDQEP($\mathscr{A} \cup \{j\}$) using B^+, G^+ and $((\Psi^+)^{-1})_{1,1}$, and obtain the step direction from the difference between this solution and the current variables. In the following section we show how this can be done and prove that we can still guarantee finite termination of the algorithm. In Section 4.2, we present the complete dual-active-set method for positive semi-definite quadratic programming and illustrate its operation using a small example. Finally, we indicate how one might proceed towards a numerically stable implementation of the method.

4.1. The look-ahead method for deactivation

From results in Section 3.2, it can be seen that if $(\bar{x}, \bar{y})$ and $\bar{\boldsymbol{\lambda}}$ solve the SDQEP($\mathscr{A} \cup \{j\}$) then

$$\begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix} = (B^+)^{\mathrm{T}} \boldsymbol{b}_{\mathscr{A} \cup \{j\}} - G^+ \begin{pmatrix} \boldsymbol{p} \\ \boldsymbol{\rho} \end{pmatrix} \tag{39}$$

and

$$\bar{\boldsymbol{\lambda}} = B^+ \nabla f(\bar{x}, \bar{y}).$$

If (x, y) and $\boldsymbol{\lambda}^+$ are the current variables, we choose the step directions

$$\begin{pmatrix} z \\ w \end{pmatrix} = \frac{1}{t_1} \left\{ \begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix} - \begin{pmatrix} x \\ y \end{pmatrix} \right\}$$

and

$$\boldsymbol{r} = -\frac{1}{t_1} \left(\bar{\boldsymbol{\lambda}}_{\setminus \{j\}} - \boldsymbol{\lambda}^+_{\setminus \{j\}} \right)$$

where

$$t_1 = \bar{\lambda}_j - \lambda_j^+ . \tag{40}$$

(Below we give efficient formulae for calculating these directions.) The step length t_2 can be calculated as before from $\boldsymbol{r}$ and $\boldsymbol{\lambda}^+$. Using these step directions and step lengths, the algorithm can proceed as before. We will prove that the same results hold for these step directions as hold for the original directions, and so guarantee finite termination of the modified "look-ahead" algorithm. Note that during an activation, the "look-ahead" method only comes into effect after a deactivation, so we must have $((\boldsymbol{x}, \boldsymbol{y}), \mathscr{A}, j)$ a V-triple by the theorems in the previous section.

Lemma 3. *The step length t_1 is non-negative, and if t_1 is zero then $\bar{\boldsymbol{\lambda}}$ is feasible, i.e. $\bar{\boldsymbol{\lambda}} \geqslant \mathbf{0}$.*

Proof. B^+ is defined in terms of two submatrices, which we will call B_1^+ and B_2^+, i.e. $B^+ = (B_1^+ \; B_2^+)$. Similarly, G^+ is defined in terms of three submatrices, which we will call $G_{1,1}^+$, $G_{1,2}^+$ and $G_{2,2}^+$, i.e.

$$G^+ = \begin{pmatrix} G_{1,1}^+ & G_{1,2}^+ \\ (G_{1,2}^+)^{\mathrm{T}} & G_{2,2}^+ \end{pmatrix}.$$

Using these submatrices, we have that

$$\bar{\boldsymbol{x}} = (B_1^+)^{\mathrm{T}} \boldsymbol{b}_{\mathscr{A} \cup \{j\}} - G_{1,1}^+ \boldsymbol{p} - G_{1,2}^+ \boldsymbol{\rho},$$

and

$$\bar{\boldsymbol{y}} = (B_2^+)^{\mathrm{T}} \boldsymbol{b}_{\mathscr{A} \cup \{j\}} - (G_{1,2}^+)^{\mathrm{T}} \boldsymbol{p} - G_{2,2}^+ \boldsymbol{\rho}$$

from (39). We can find a similar form for $\boldsymbol{x}$ and $\boldsymbol{y}$ from the definition of a V-triple:

$$\boldsymbol{x} = (B_1^+)^{\mathrm{T}} \begin{pmatrix} \boldsymbol{b}_{\mathscr{A}} \\ \beta \end{pmatrix} - G_{1,1}^+ \boldsymbol{p} - G_{1,2}^+ \boldsymbol{\rho},$$

and

$$\boldsymbol{y} = (B_2^+)^{\mathrm{T}} \begin{pmatrix} \boldsymbol{b}_{\mathscr{A}} \\ \beta \end{pmatrix} - (G_{1,2}^+)^{\mathrm{T}} \boldsymbol{p} - G_{2,2}^+ \boldsymbol{\rho}$$

where

$$\beta = \boldsymbol{c}_j^{\mathrm{T}} \boldsymbol{x} + \boldsymbol{d}_j^{\mathrm{T}} \boldsymbol{y}.$$

Hence

$$\bar{\boldsymbol{x}} - \boldsymbol{x} = (B_1^+)^{\mathrm{T}} \begin{pmatrix} \mathbf{0} \\ \delta \end{pmatrix} \tag{41}$$

where

$$\delta = -s_j(\boldsymbol{x}, \boldsymbol{y}) > 0.$$

Now

$$\begin{aligned} \bar{\boldsymbol{\lambda}} - \boldsymbol{\lambda}^+ &= B^+ \nabla f(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}}) - B^+ \nabla f(\boldsymbol{x}, \boldsymbol{y}) \\ &= B^+ \begin{pmatrix} Q(\bar{\boldsymbol{x}} - \boldsymbol{x}) \\ \mathbf{0} \end{pmatrix} = B_1^+ Q (B_1^+)^{\mathrm{T}} \begin{pmatrix} \mathbf{0} \\ \delta \end{pmatrix} \end{aligned}$$

by (41), so if γ is the jth row vector in B_1^+, then

$$t_1 = \bar{\lambda}_j - \lambda_j^+ = \delta\gamma Q\gamma^{\mathrm{T}} \geqslant 0$$

since Q is positive definite and $\delta > 0$. □

The results developed in the above lemma show us how we may efficiently update the primal and dual variables in "look-ahead" mode. From above, using B, G and $(\Psi^{-1})_{1,1}$ updated *after* j has been added to the active set, we see that

$$\bar{\boldsymbol{\lambda}} - \boldsymbol{\lambda}^+ = -s_j(x, y)(\Psi^{-1})_{1,1}\boldsymbol{e}_j$$

since $(\Psi^{-1})_{1,1} = B_1 Q(B_1)^{\mathrm{T}}$, and

$$\begin{pmatrix} \bar{x} \\ \bar{y} \end{pmatrix} - \begin{pmatrix} x \\ y \end{pmatrix} = -s_j(x, y)\begin{pmatrix} B_1^{\mathrm{T}}\boldsymbol{e}_j \\ B_2^{\mathrm{T}}\boldsymbol{e}_j \end{pmatrix}$$

where $\boldsymbol{e}_j \in \mathbb{R}^{|\mathscr{A}|}$ is the unit vector for constraint j. Efficient factorizations of B and $(\Psi^{-1})_{1,1}$ are given in Section 4.3.

If we are given a V-triple $((x, y), \mathscr{A}, j)$, and calculate $t_1 = 0$ from (40) then since $(\bar{x}, \bar{y})$ solves the SDQEP($\mathscr{A} \cup \{j\}$) (by definition) and $\bar{\boldsymbol{\lambda}} \geqslant \boldsymbol{0}$ (by the above lemma), we can add j to $\mathscr{A}$ and set $(\bar{x}, \bar{y})$ and $\bar{\boldsymbol{\lambda}}$ to be the new variables. In the case that $t_1 > 0$, we obtain new variables as described in the following theorem, which parallels Theorem 1.

Theorem 4. *Given $((x, y), \mathscr{A}, j)$, a V-triple, and step directions and step lengths as defined above, with $t_1 > 0$, if $\boldsymbol{\lambda}^+ - t_1\boldsymbol{r} \geqslant \boldsymbol{0}$ set $s = t_1$; otherwise set $s = t_2$ and set k to be the constraint chosen to deactivate. Then*

$$s_j(\hat{x}, \hat{y}) \geqslant s_j(x, y)$$

and

$$f(\hat{x}, \hat{y}) \geqslant f(x, y)$$

where

$$\begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix} = \begin{pmatrix} x \\ y \end{pmatrix} + s\begin{pmatrix} z \\ w \end{pmatrix}.$$

Furthermore, if $s = t_2$ then $((\hat{x}, \hat{y}), \mathscr{A}\setminus\{k\}, j)$ is a V-triple. In addition, if $s = t_1$ then $(\hat{x}, \hat{y})$ solves the SDQEP($\mathscr{A} \cup \{j\}$).

Proof. Since $s_j(\bar{x}, \bar{y}) = 0$ we have

$$s_j(\hat{x}, \hat{y}) = \left(1 - \frac{s}{t_1}\right)s_j(x, y), \tag{42}$$

so because $s \geqslant 0$ and $s_j(x, y) < 0$, it must be that $s_j(\hat{x}, \hat{y}) \geqslant s_j(x, y)$. Now

$$f(\hat{x}, \hat{y}) - f(x, y) = s(z^{\mathrm{T}} \quad w^{\mathrm{T}})\nabla f(x, y) + \tfrac{1}{2}s^2 z^{\mathrm{T}}Qz$$

and

$$\nabla f(x, y) = \begin{pmatrix} N^+ \\ M^+ \end{pmatrix} \boldsymbol{\lambda}^+$$

from the properties of a V-triple. Also

$$(z^{\mathrm{T}} \;\; w^{\mathrm{T}}) \begin{pmatrix} N^+ \\ M^+ \end{pmatrix} = \frac{1}{t_1} (\mathbf{0}^{\mathrm{T}} \;\; \delta)$$

and hence

$$f(\hat{x}, \hat{y}) - f(x, y) = \frac{s}{t_1} \delta \lambda_j^+ + \tfrac{1}{2} s^2 z^{\mathrm{T}} Q z$$

which is non-negative, since t_1, $\delta > 0$, s, $\lambda_j^+ \geqslant 0$, and Q is positive definite.

To prove that if $s = t_2$ then $((\hat{x}, \hat{y}), \mathscr{A} \setminus \{k\}, j)$ is a V-triple, we firstly note that since $t_2 < t_1$, $s_j(\hat{x}, \hat{y}) < 0$ by (42). It is also obvious that $s_i(\hat{x}, \hat{y}) = 0$ for all $i \in \mathscr{A} \setminus \{k\}$. The following can be proved directly from definitions:

$$\begin{pmatrix} Qz \\ \mathbf{0} \end{pmatrix} = \begin{pmatrix} N^+ \\ M^+ \end{pmatrix} \begin{pmatrix} -\boldsymbol{r} \\ 1 \end{pmatrix},$$

and

$$\begin{aligned} \nabla f(\hat{x}, \hat{y}) &= \nabla f(x, y) + t_2 \begin{pmatrix} Qz \\ \mathbf{0} \end{pmatrix} \\ &= \begin{pmatrix} N^+ \\ M^+ \end{pmatrix} \left\{ \boldsymbol{\lambda}^+ + t_2 \begin{pmatrix} -\boldsymbol{r} \\ 1 \end{pmatrix} \right\}. \end{aligned}$$

Hence

$$\begin{aligned} G^+_{\mathscr{A} \setminus \{k\}} \nabla f(\hat{x}, \hat{y}) &= (\lambda_k^+ - t_2 r_k) G^+_{\mathscr{A} \setminus \{k\}} \begin{pmatrix} \boldsymbol{c}_k \\ \boldsymbol{d}_k \end{pmatrix} \qquad \text{by (6)} \\ &= \mathbf{0} \end{aligned}$$

from the definition of t_2 and k. Also

$$\begin{aligned} B^+_{\mathscr{A} \setminus \{k\}} \nabla f(\hat{x}, \hat{y}) &= \boldsymbol{\lambda}^+_{\setminus \{k\}} + t_2 \begin{pmatrix} -\boldsymbol{r}_{\setminus \{k\}} \\ 1 \end{pmatrix} + (\lambda_k^+ - t_2 r_k) B^+_{\mathscr{A} \setminus \{k\}} \begin{pmatrix} \boldsymbol{c}_k \\ \boldsymbol{d}_k \end{pmatrix} \qquad \text{by (7)} \\ &= \boldsymbol{\lambda}^+_{\setminus \{k\}} + t_2 \begin{pmatrix} -\boldsymbol{r}_{\setminus \{k\}} \\ 1 \end{pmatrix} \qquad \text{by the definition of } k \\ &\geqslant \mathbf{0} \end{aligned}$$

from the definition of t_2. This completes the proof that $((\hat{x}, \hat{y}), \mathscr{A} \setminus \{k\}, j)$ is a V-triple.

Now if $s = t_1$ we have that $(\hat{x}, \hat{y})$ is just $(\bar{x}, \bar{y})$, which is defined to be the solution of the SDQEP($\mathscr{A} \cup \{j\}$). □

This theorem, together with the theorems in Section 3.4, furnish proof of finite termination of the dual-active-set method with look-ahead deactivation which we present in the section below.

4.2. The dual-active-set algorithm

In this section, we present the dual-active-set-method with look-ahead deactivation for solving positive semi-definite quadratic programming problems. The theorems given in Sections 3.4 and 4.1 prove finite termination of this algorithm. An implementation issue with this algorithm is the selection of the primal infeasible constraint to activate. Like Goldfarb and Idnani, we suggest selecting the maximally primal infeasible constraint, i.e. the constraint with the most negative $s_j(x, y)$. Results for network optimization problems given in [2] as well as those of Goldfarb and Idnani for the positive definite case, testify to the efficacy of this strategy.

4.2.1. The algorithm

The dual-active-set method with look-ahead deactivation is presented below.

Find initial $\mathscr{A}$ as described in Section 3.3 and calculate B, G and $(\Psi^{-1})_{1,1}$.

Set $\binom{x}{y} \leftarrow \binom{B_1^{\mathrm{T}} b_{\mathscr{A}}}{B_2^{\mathrm{T}} b_{\mathscr{A}}} - G\binom{p}{\rho}$ and $\boldsymbol{\lambda} \leftarrow (\Psi^{-1})_{1,1} \boldsymbol{b}_{\mathscr{A}} + B\binom{p}{\rho}$.

while $\exists j$ with $s_j(x, y) < 0$ **do**

Set $\lambda \leftarrow \binom{\lambda}{0}$, lookahead $\leftarrow$ FALSE and deactivating $\leftarrow$ FALSE.

Set $\binom{z}{w} \leftarrow G\binom{c_j}{d_j}$ and $\boldsymbol{r} \leftarrow B\binom{c_j}{d_j}$.

if $\binom{z}{w} \neq \mathbf{0}$ **then** set $t_1 \leftarrow -s_j(x, y)/(c_j^{\mathrm{T}} z + d_j^{\mathrm{T}} w)$

else if $\boldsymbol{r} \leqslant 0$ **then** STOP (the problem is infeasible)

if $\binom{z}{w} = \mathbf{0}$ or $\boldsymbol{\lambda} + t_1\binom{-r}{1} \ngeqslant \mathbf{0}$ **then** deactivating $\leftarrow$ TRUE.

while deactivating = TRUE **do**

Set $t_2 \leftarrow \min_{i \in \mathscr{A},\, r_i > 0}\{\lambda_i / r_i\}$ and set $k \in \mathscr{A}$ to be the constraint which achieves this minimum.

Set $\binom{x}{y} \leftarrow \binom{x}{y} + t_2\binom{z}{w}$, $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda}_{\setminus\{k\}} + t_2\binom{-r}{1}_{\setminus\{k\}}$, and $\mathscr{A} \leftarrow \mathscr{A} \setminus \{k\}$.

if rank $(D_{\mathscr{A}}) < l$ **then**

Set lookahead $\leftarrow$ TRUE and $\mathscr{A} \leftarrow \mathscr{A} \cup \{j\}$.

Update B, G and $(\Psi^{-1})_{1,1}$ to reflect the exchange of constraints k and j in the active set.

else if lookahead = FALSE **then**

Update B, G and $(\Psi^{-1})_{1,1}$ to reflect the removal of constraint k from the active set.

Set $\binom{z}{w} \leftarrow G\binom{c_j}{d_j}$ and $r \leftarrow B\binom{c_j}{d_j}$.

if $\binom{z}{w} \neq \mathbf{0}$ **then**

Set $t_1 \leftarrow -s_j(x, y)/(c_j^{\mathrm{T}} z + d_j^{\mathrm{T}} w)$.

if $\boldsymbol{\lambda} + t_1\binom{-r}{1} \geqslant \mathbf{0}$ **then** deactivating $\leftarrow$ FALSE.

else if $\boldsymbol{r} \leqslant \mathbf{0}$ **then** STOP (the problem is infeasible).

else

Update B, G and $(\Psi^{-1})_{1,1}$ to reflect the removal of constraint k from the active set.

endif

if lookahead = TRUE **then**

Set $\boldsymbol{\mu} \leftarrow -s_j(\boldsymbol{x}, \boldsymbol{y})(\Psi^{-1})_{1,1}e_j$ and $t_1 \leftarrow \mu_j$.
if $t_1 > 0$ **then**
Set $\binom{z}{w} \leftarrow -s_j(\boldsymbol{x}, \boldsymbol{y})/t_1\binom{B_1^{\mathrm{T}}e_j}{B_2^{\mathrm{T}}e_j}$ and $\boldsymbol{r} \leftarrow -(1/t_1)\boldsymbol{\mu}_{\setminus\{j\}}$.
else ($t_1 = 0$)
Set $\binom{x}{y} \leftarrow \binom{x}{y} - s_j(\boldsymbol{x}, \boldsymbol{y})\binom{B_1^{\mathrm{T}}e_j}{B_2^{\mathrm{T}}e_j}$ and $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} + \boldsymbol{\mu}$.
endif
if $t_1 = 0$ or $\boldsymbol{\lambda} + t_1\binom{-r}{1} \geqslant \mathbf{0}$ **then** deactivating $\leftarrow$ FALSE.
endif
enddo
if $t_1 > 0$ **then** set $\binom{x}{y} \leftarrow \binom{x}{y} + t_1\binom{z}{w}$ and $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} + t_1\binom{-r}{1}$.
if lookahead = FALSE **then**
Set $\mathscr{A} \leftarrow \mathscr{A} \cup \{j\}$ and update B, G and $(\Psi^{-1})_{1,1}$ to reflect the addition of j to the active set.
endif
enddo

Notice that if it is desired, the look-ahead method could be used throughout the algorithm, except in the event that the constraint to be added to the active set is linearly dependent on the already active constraints.

4.2.2. A sample problem

We will illustrate the operation of the positive semi-definite dual-active-set algorithm using the following example:

$$\begin{array}{ll} \text{min} & x_1^2 + x_2 \\ \text{subject to} & x_1 + x_2 \geqslant 1, \quad \text{and} \\ & x_1 + 2x_2 \geqslant 2. \end{array}$$

The feasible region and lines of constant objective function for this problem are shown in Fig. 1. Clearly the solution for this problem is at $x_1 = 0.25$ and $x_2 = 0.875$. The minimum value is 0.9375. The values of the current primal variables at each stage of the algorithm are indicated in Fig. 1.

Consider how the algorithm would operate on this problem if the initial active set is $\mathscr{A} = \{1\}$. B, G and $(\Psi^{-1})_{1,1}$ are calculated:

$$B = (0 \;\; 1), \qquad G = \begin{pmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{pmatrix}, \quad \text{and} \quad (\Psi^{-1})_{1,1} = (0).$$

Primal and dual variables are computed:

$$\begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \\ \frac{1}{2} \end{pmatrix} \quad \text{and} \quad \boldsymbol{\lambda} = (1).$$

Now $s_2(\boldsymbol{x}, \boldsymbol{y}) = -\frac{1}{2} < 0$ so $j = 2$ and we set

$$\boldsymbol{\lambda} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

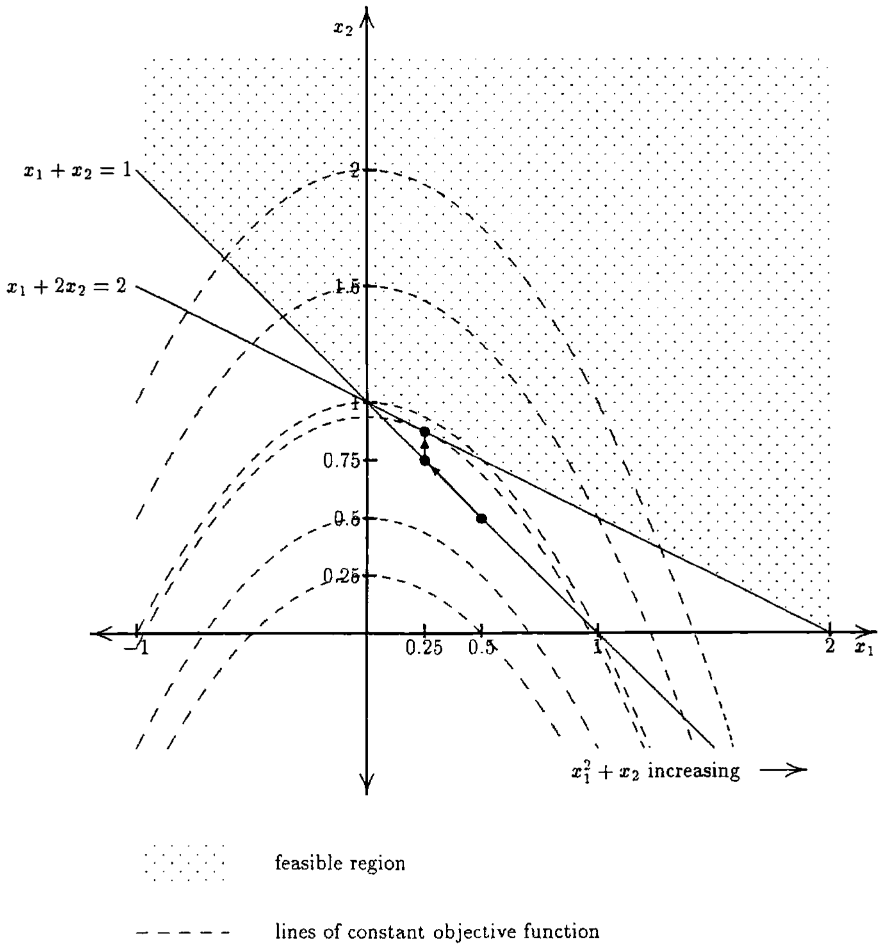

Fig. 1. The feasible region and lines of constant objective function for the sample problem. The primal variables at each step of the algorithm are indicated.

Also *lookahead* is set to FALSE,

$$\begin{pmatrix} z \\ w \end{pmatrix} = \begin{pmatrix} -\frac{1}{2} \\ \frac{1}{2} \end{pmatrix} \quad \text{and} \quad r = (2).$$

Since $(z\ w) \neq \mathbf{0}$ we set $t_1 = -s_2(x, y)/(c_j^{\mathrm{T}} z + d_j^{\mathrm{T}} w) = 1$. Now $t_1 > 0$ and

$$\boldsymbol{\lambda} + t_1 \begin{pmatrix} -r \\ 1 \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \end{pmatrix} \not\geq \mathbf{0}$$

so it must be that some dual variable will become infeasible as a result of making constraint 2 active. To determine how far we should move in the current direction we compute

$$t_2 = \min_{i \in \mathscr{A},\ r_i > 0} \left\{ \frac{\lambda_i}{r_i} \right\} = \frac{1}{2}$$

and set $k=1$. Moving the primal and dual variables in the current direction, and removing k from the current active set, we get

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \frac{1}{4} \\ \frac{3}{4} \end{pmatrix} \quad \text{and} \quad \boldsymbol{\lambda} = \begin{pmatrix} \frac{1}{2} \end{pmatrix}$$

and $\mathscr{A} = \emptyset$. Now the rank of $D_{\mathscr{A}}$ is 0 which is less that $l=1$, so we set *lookahead* to be TRUE, add j to $\mathscr{A}$ so that $\mathscr{A} = \{2\}$, and update B, G and $(\Psi^{-1})_{1,1}$:

$$B = \begin{pmatrix} 0 & \frac{1}{2} \end{pmatrix}, \qquad G = \begin{pmatrix} \frac{1}{2} & -\frac{1}{4} \\ -\frac{1}{4} & \frac{1}{8} \end{pmatrix}, \quad \text{and} \quad (\Psi^{-1})_{1,1} = (0).$$

Since *lookahead* is now TRUE, we calculate $\boldsymbol{\mu} = (0)$ and $t_1 = 0$. Noting that $s_2(\frac{1}{4}, \frac{3}{4}) = -\frac{1}{4}$, we set the primal and dual variables to be:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \frac{1}{4} \\ \frac{3}{4} \end{pmatrix} + \frac{1}{4} \begin{pmatrix} B_1^{\mathrm{T}}(1) \\ B_2^{\mathrm{T}}(1) \end{pmatrix} = \begin{pmatrix} \frac{1}{4} \\ \frac{7}{8} \end{pmatrix} \quad \text{and} \quad \boldsymbol{\lambda} = \begin{pmatrix} \frac{1}{2} \end{pmatrix}.$$

This solution is primal feasible, so the algorithm terminates with the optimal solution.

4.3. Towards a numerically stable implementation

Our primary purpose in this paper is to show that, in theory, the Goldfarb and Idnani dual-active-set method can be generalized to the positive semi-definite case. However we feel that some discussion of the issues involved in a practical implementation of the method is warranted. This discussion is not intended to be complete or exhaustive; in fact we expect there to be many ways of effecting a practical implementation of the method we have presented. We include the material in this section only to indicate that it is possible to express the necessary matrices by way of numerically stable factorizations. We do not intend that this be the most efficient or effective representation.

The method we presented in the previous section relies on the complicated matrices B and G to compute the search directions. In practice, these matrices are unlikely to be useful as computational tools. However we can express them in terms of matrix factors obtained from numerically stable factorizations which are analagous to those described by Goldfarb and Idnani. These factorizations are based on a Cholesky factorization

$$Q = LL^{\mathrm{T}} \tag{43}$$

of the positive definite symmetric matrix Q, a QR factorization

$$T = \overline{Q}\begin{pmatrix} \overline{R} \\ 0 \end{pmatrix} = \begin{pmatrix} \overline{Q}_{11} & \overline{Q}_{12} \\ \overline{Q}_{21} & \overline{Q}_{22} \end{pmatrix} \begin{pmatrix} \overline{R} \\ 0 \end{pmatrix} \tag{44}$$

of the $(l+q) \times m$ matrix

$$T = \begin{pmatrix} L^{-1}N \\ M \end{pmatrix} \tag{45}$$

and a QR factorization

$$U = \hat{Q}\begin{pmatrix} \hat{R} \\ 0 \end{pmatrix} = \begin{pmatrix} \hat{Q}_1 & \hat{Q}_2 \end{pmatrix} \begin{pmatrix} \hat{R} \\ 0 \end{pmatrix} \tag{46}$$

of the $q \times m$ matrix

$$U = \bar{R}^{-T} M^{T} = \bar{Q}_{21}^{T}. \tag{47}$$

L is a $q \times q$ lower triangular matrix, $\bar{R}$ is an $m \times m$ upper triangular matrix, $\bar{Q}$ is a $(q+l) \times (q+l)$ orthogonal matrix, partitioned so that $\bar{Q}_{11}$ has q rows and m columns, $\hat{R}$ is an $l \times l$ upper triangular matrix, and $\hat{Q}$ is an $m \times m$ orthogonal matrix, partitioned so that $\hat{Q}_1$ has l columns, where $m \stackrel{\text{def}}{=} |\mathscr{A}|$. Using (43)–(47), we obtain

$$R = \bar{R}^{-1}\bar{R}^{-T} \quad \text{and} \quad S = \hat{R}^{-1}\hat{R}^{-T},$$

and hence that

$$B = \begin{pmatrix} \bar{R}^{-1}\hat{Q}_2\hat{Q}_2^{T}\bar{Q}_{11}^{T}L^{-1} & \bar{R}^{-1}\hat{Q}_1\hat{R}^{-T} \end{pmatrix},$$

$$G = \begin{pmatrix} L^{-T}\left(I_q - \bar{Q}_{11}\hat{Q}_2\hat{Q}_2^{T}\bar{Q}_{11}^{T}\right)L^{-1} & -L^{-T}\bar{Q}_{11}\hat{Q}_1\hat{R}^{-T} \\ -\hat{R}^{-1}\hat{Q}_1^{T}\bar{Q}_{11}^{T}L^{-1} & \hat{R}^{-1}\hat{R}^{-T} - I_l \end{pmatrix}$$

and

$$(\Psi^{-1})_{1,1} = \bar{R}^{-1}\hat{Q}_2\hat{Q}_2^{T}\bar{R}^{-T}.$$

This can be simplified slightly if we store and maintain $V_1 =^{\text{def}} L^{-T}\bar{Q}_{11}$ and $V_2 =^{\text{def}} \hat{R}^{-1}\hat{Q}_1^{T}$ rather than $\bar{Q}_{11}$ and $\hat{Q}_1$, since $\bar{Q}_{11}$ only ever appears in conjunction with L^{-T} and $\hat{Q}_1^{T}$ only ever appears in conjunction with $\hat{R}^{-1}$.

4.4. Relationship with a primal active set method applied to the dual problem

As Fletcher [9] observes, the dual iterates produced by the Goldfarb and Idnani method are identical to those that would be produced by applying a primal active set method (such as those described in Chapter 10.3 of [9] or in [11]), to the dual problem. The key differences are in the implementation.

Difficulties caused to a primal active set method by degeneracy of the constraints in the equality problem are avoided in the Goldfarb and Idnani method because the only constraints in the dual problem are the nonnegativity constraints; these cannot lead to degeneracy. This property would appear to fail in the case of the semi-definite problem we wish to solve, where the dual constraints are $D\boldsymbol{\lambda} = \boldsymbol{\rho}$ and $\boldsymbol{\lambda} \geqslant \mathbf{0}$. In this case the equality problem solved by a primal active set method to determine the dual step $\boldsymbol{\mu}$ would be

$$\begin{aligned} \min \quad & \tfrac{1}{2}\boldsymbol{\mu}^{T}(C^{T}Q^{-1}C)\boldsymbol{\mu} - \left(C^{T}Qp + \boldsymbol{b}\right)^{T}\boldsymbol{\mu} \\ \text{subject to} \quad & D\boldsymbol{\mu} = \mathbf{0}, \\ & \mu_i = 0 \quad \forall i \notin \mathscr{A} \end{aligned}$$

which is equivalent to

$$\begin{aligned} \min \quad & \tfrac{1}{2}\boldsymbol{\mu}_{\mathscr{A}}^{T}(N^{T}Q^{-1}N)\boldsymbol{\mu}_{\mathscr{A}} - \left(N^{T}Qp + \boldsymbol{b}_{\mathscr{A}}\right)^{T}\boldsymbol{\mu}_{\mathscr{A}} \\ \text{subject to} \quad & M\boldsymbol{\mu}_{\mathscr{A}} = \mathbf{0}. \end{aligned}$$

It is not obvious that these constraints could not become degenerate. However, noting that a primal active set method applied to the dual problem is always in "look-ahead" mode, we see that Lemma 2 in effect proves exactly that: degeneracy of the constraints in the dual equality problem is still avoided.

We observe that the algorithm we have presented preserves the property that the dual iterates produced correspond to those that would be produced by a primal active set method applied to the dual problem. Again, the benefits of the implementation we present are, in comparison, commensurate with those of the Goldfarb and Idnani method.

5. Conclusions

We have presented a very natural extension of the Goldfarb and Idnani dual-active-set algorithm for positive definite quadratic programming which enables positive semi-definite problems to be solved. Matrices analogous to those used by Goldfarb and Idnani for determining search directions were defined, and new initial conditions for the algorithm were developed. These conditions meant that there needed to be, in some sense, a "lower bound" on the active set, as well as the "upper bound" already encountered in the original method. In the original method, (and also in this one), the constraints in the active set must be linearly independent; our new conditions imply that the constraints in the active set must span a certain subspace. In order to maintain this condition, we introduce a new method of obtaining search directions, the "look-ahead" method. We have proved in this paper that the resulting algorithm will terminate in a finite number of iterations with the solution to the positive semi-definite programming problem. Although we do require that our problem be in a special form, we have shown that this not a restrictive requirement. In addition, we have indicated that the matrices we use to determine search directions are amenable to numerically stable implementation.

Acknowledgements

We thank H. Wolkowicz of the Department of Combinatorics and Optimization at the University of Waterloo for his helpful suggestions on issues of numerical stability and practical implementation. We are also grateful to W. Murray and an anonymous referee for their careful reading and detailed, insightful comments.

References

[1] N. Boland, C.J. Goh and A.I. Mees, An algorithm for solving quadratic network flow problems, *Applied Mathematics Letters* 4 (1991) 61–64.

[2] N. Boland, C.J. Goh and A.I. Mees, An algorithm for non-linear network programming: implementation, results and comparisons, *Journal of the Operational Research Society* 43 (1992) 979–992.

[3] N. Boland, Some problems in network optimization and quadratic programming, PhD Thesis, Department of Mathematics, University of Western Australia (1992).

[4] D. Burton and Ph.L. Toint, On an instance of the inverse shortest paths problem, *Mathematical Programming* 53 (1992) 45–61.

[5] A.R. Conn, N.I.M. Gould, M. Lescrenier and Ph.L. Toint, Performance of a multifrontal scheme for partially separable optimization, Research Report CS-88-04, Computer Science Department, University of Waterloo (1988).

[6] A.R. Conn, N.I.M. Gould and Ph.L. Toint, An introduction to the structure of large scale nonlinear optimization problems and the LANCELOT project, Research Report CS-89-63, Computer Science Department, University of Waterloo (1989).

[7] A.R. Conn, N. Gould and Ph.L. Toint, Large-scale nonlinear constrained optimization, Research Report RAL-92-066, Rutherford Appleton Laboratory (1992).

[8] J. Demmel, Jacobis method is more accurate than QR, *SIAM Journal on Matrix Analysis and Applications* 13 (1992) 1204–1245.

[9] R. Fletcher, *Practical Methods of Optimization* (Wiley, Chichester, 1987).

[10] A.T. Ernst, X.Q. Yang and N.L. Boland, Exterior point methods for quadratic cost multicommodity flow, submitted for publication, 1996.

[11] P.E. Gill and W. Murray, Numerically stable methods for quadratic programming, *Mathematical Programming* 14 (1978) 349–372.

[12] P.E. Gill, W. Murray and M. Wright, *Practical Optimization* (Academic Press, London, 1981).

[13] P.E. Gill, W. Murray and M. Wright, *Numerical Linear Algebra and Optimization* (Addison-Wesley, Reading, MA, 1991).

[14] D. Goldfarb and A. Idnani, A numerically stable dual method for solving strictly convex quadratic programs, *Mathematical Programming* 27 (1983) 1–33.

[15] G.H. Golub and C.F. Van Loan, *Matrix Computations* (Johns Hopkins University Press, Baltimore, MD, 1989).

[16] S.P. Han, Superlinearly convergent variable metric algorithms for general nonlinear programming problems, *Mathematical Programming* 11 (1976) 263–282.

[17] S.P. Han, A globally convergent method for nonlinear programming, *Journal of Optimization Theory and Applications* 22 (1977) 297–309.

[18] Y.Y. Lin and J.S. Pang, Iterative methods for large convex quadratic programs: a survey, *SIAM Journal on Control and Optimization* 25 (1987) 383–411.

[19] W. Murray and F.J. Prieto, A sequential quadratic programming algorithm using incomplete solution of the subproblem, Technical Report SOL90-12, Operations Research Department, Stanford University (1990).

[20] B.N. Parlett, *The Symmetric Eigenvalue Problem* (Prentice-Hall, Englewood Cliffs, NJ, 1980).

[21] M.J.D. Powell, A fast algorithm for nonlinearly constrained optimization calculations, in: G.A. Watson, ed., *Numerical analysis Dundee 1977*, Lecture Notes in Mathematics 630 (Springer, Berlin, 1977) 144–157.

[22] M.J.D. Powell, The convergence of variable metric methods for nonlinearly constrained optimization calculations, in: O.L. Magasarian, R.R. Meyer and S.M. Robinson, eds., *Nonlinear Programming 3* (Academic Press, New York, 1978) 27–63.

[23] M.J.D. Powell, Variable metric methods for constrained optimization, in: A. Bachen, M. Grotschel and B. Korte, eds., *Mathematical Programming: The State of the Art* (Springer, Berlin, 1983) 288–311.

[24] M.J.D. Powell, ZQPCVX: a Fortran subroutine for convex quadratic programming, Report DAMTP/1983/NA17, University of Cambridge (1983).

[25] M.J.D. Powell, On the quadratic programming algorithm of Goldfarb and Idnani, *Mathematical Programming Study* 25 (1985) 46–61.

[26] K. Schittkowski, The nonlinear programming method of Wilson, Han, and Powell with an augmented Lagrangian type line search function, Part I: convergence analysis, *Numerische Mathematik* 38 (1981) 83–114.

[27] K. Schittkowski, On the convergence of a sequential quadratic programming method with an augmented Lagrangian line search function, *Mathematische Operationsforschung und Statistik, Ser. Optimization* 14 (1983) 197–216.

[28] J. Stoer, A dual algorithm for solving degenerate linearly constrained linear least squares problems, *Journal of Numerical Linear Algebra with Applications* 1 (1992) 103–131.

[29] Ph.L. Toint and D. Tuyttens, On large scale nonlinear network optimization, *Mathematical Programming* 48 (1990) 125–159.

[30] R.B. Wilson, A simplicial algorithm for concave programming, Ph.D. Dissertation, Graduate School of Business Administration, Harvard University, Boston (1963).